Report | HW 4

CS663 - Digital Image Processing

Omkar Shirpure (22B0910)
Suryansh Patidar (22B1036)
Aditya Saran (22B1839)

Contents

1 Q1

(a) Prove that the covariance matrix in PCA is symmetric and positive semi-definite.

1. **Covariance matrix is symmetric**: The covariance matrix Σ is defined as:

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(x_i - \bar{x})^T$$

where $x_i \in \mathbb{R}^d$ are the data points, and $\bar{x}$ is the mean vector. The covariance matrix is symmetric because:

$$\Sigma^T = \left(\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(x_i - \bar{x})^T\right)^T = \frac{1}{N}\sum_{i=1}^{N}\left((x_i - \bar{x})(x_i - \bar{x})^T\right)^T = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(x_i - \bar{x})^T = \Sigma$$

Thus, $\Sigma^T = \Sigma$, so the covariance matrix is symmetric.

2. **Covariance matrix is positive semi-definite**: A symmetric matrix is positive semi-definite if for any non-zero vector $v \in \mathbb{R}^d$, the following holds:

$$v^T \Sigma v \geq 0$$

Substituting the definition of Σ:

$$v^T \Sigma v = v^T\left(\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(x_i - \bar{x})^T\right)v = \frac{1}{N}\sum_{i=1}^{N} v^T(x_i - \bar{x})(x_i - \bar{x})^T v = \frac{1}{N}\sum_{i=1}^{N}(v^T(x_i - \bar{x}))^2$$

Since $(v^T(x_i - \bar{x}))^2 \geq 0$ for all i, it follows that $v^T \Sigma v \geq 0$, meaning that the covariance matrix is positive semi-definite.

(b) Prove that the eigenvectors of a symmetric matrix are orthonormal.

Let v_i and v_j be eigenvectors corresponding to distinct eigenvalues λ_i and λ_j, respectively. We need to show that $v_i^T v_j = 0$ (i.e., they are orthogonal).

Since $Av_i = \lambda_i v_i$ and $Av_j = \lambda_j v_j$, consider the following:

$$v_i^T A v_j = v_i^T \lambda_j v_j = \lambda_j v_i^T v_j$$

On the other hand, since A is symmetric, $(Av_i)^T = v_i^T A$, so:

$$v_i^T A v_j = (Av_i)^T v_j = \lambda_i v_i^T v_j$$

Equating both expressions:

$$\lambda_i v_i^T v_j = \lambda_j v_i^T v_j$$

Since $\lambda_i \neq \lambda_j$, it must be that $v_i^T v_j = 0$, proving that eigenvectors corresponding to distinct eigenvalues are orthogonal.

Furthermore, if the eigenvectors are normalized (i.e., $v_i^T v_i = 1$), they are orthonormal.

(c) Error in approximating x_i using only large eigenvalues.

Given that x_i is approximated as:

$$\tilde{x}_i = \bar{x} + \sum_{l=1}^{k} V_l \alpha_{il}$$

the error in this approximation is the squared difference:

$$\text{Error} = \frac{1}{N} \sum_{i=1}^{N} \|\tilde{x}_i - x_i\|_2^2$$

$$\|\tilde{x}_i - x_i\|_2^2 = \left\| \sum_{l=1}^{k} V_l \alpha_{il} - \sum_{l=1}^{d} V_l \alpha_{il} \right\|_2^2$$

$$= \left\| \sum_{l=k+1}^{d} V_l \alpha_{il} \right\|^2$$

Now, since the eigenvectors V_l are orthonormal, the norm of this sum can be written as the sum of the squares of the individual terms:

$$= \sum_{l=k+1}^{d} \alpha_{il}^2$$

The key point in PCA is that the variance along each eigenvector V_l is given by the corresponding eigenvalue λ_l of the covariance matrix. The squared projection α_{il}^2 represents how much of the variance in x_i is explained by the eigenvector V_l. The average squared projection across all data points is equal to the eigenvalue λ_l. Thus:

$$\frac{1}{N} \sum_{i=1}^{N} \alpha_{il}^2 = \lambda_l$$

Using this, the total error can be written as:

$$\text{Total Error} = \sum_{l=k+1}^{d} \lambda_l$$

where λ_l are the eigenvalues of the covariance matrix. Hence, the error in the approximation is the sum of the eigenvalues corresponding to the discarded dimensions.
Since we are given that the only the k eigenvalues are large and rest are very small. So the sum of the remaining eigenvalues will be small too and hence our error will be small.

(d) Principal Components of Two Uncorrelated Random Variables

1. Principal Components when X_1 and X_2 have different variances

You are given two uncorrelated zero-mean random variables X_1 and X_2. Specifically:

$$X_1 \sim \mathcal{N}(0, 100) \quad \text{(Gaussian distribution with variance 100)}$$

$$X_2 \sim \mathcal{N}(0, 1) \quad \text{(Gaussian distribution with variance 1)}$$

The covariance matrix of the vector $\mathbf{X} = [X_1, X_2]^T$ is:

$$\mathbf{C} = \begin{bmatrix} \text{Var}(X_1) & \text{Cov}(X_1, X_2) \\ \text{Cov}(X_1, X_2) & \text{Var}(X_2) \end{bmatrix}$$

Since X_1 and X_2 are uncorrelated, Cov$(X_1, X_2) = 0$, so the covariance matrix becomes:

$$\mathbf{C} = \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix}$$

To find the principal components, we need to find the eigenvalues and eigenvectors of this covariance matrix.

The eigenvalues of a diagonal matrix like **C** are just the diagonal entries, so the eigenvalues are:

$$\lambda_1 = 100 \quad \text{and} \quad \lambda_2 = 1$$

The eigenvector corresponding to $\lambda_1 = 100$ comes from solving the equation

$$Cv_1 = 100v_1$$

Taking $v_1 = [a\ b]^T$, we get

$$100a = 100a\,,\ b = 100b$$

Clearly *a* can be anything but *b=0*
So eigenvector corresponding to $\lambda_1 = 100$ is $\mathbf{v}_1 = \{[a\ 0]^T | a \in \mathbb{R}\}$ The eigenvector corresponding to $\lambda_2 = 1$ comes from solving the equation

$$Cv_2 = v_2$$

Taking $v_2 = [a\ b]^T$, we get

$$100a = a\,,\ b = b$$

Clearly *b* can be anything but *a=0*
So eigenvector corresponding to $\lambda_2 = 1$ is $\mathbf{v}_2 = \{[0\ b]^T | b \in \mathbb{R}\}$

Thus, the principal components are aligned with the axes:

- The first principal component corresponds to $\mathbf{v}_1 = [a\ 0]^T$, which captures most of the variance (along the X_1-axis).
- The second principal component corresponds to $\mathbf{v}_2 = [0\ b]^T$, capturing the smaller variance (along the X_2-axis).

2. Principal Components when X_1 and X_2 have equal variance

Now, suppose both X_1 and X_2 have the same variance. Let's say both have variance β for simplicity. Then, the covariance matrix becomes:

$$\mathbf{C} = \begin{bmatrix} \beta & 0 \\ 0 & \beta \end{bmatrix}$$

In this case:

- The eigenvalues of **C** are $\lambda_1 = \beta$ and $\lambda_2 = \beta$ (since both variances are equal).
- The covariance matrix is already diagonal and has identical eigenvalues, so any orthonormal set of vectors in $\mathbb{R}^2$ could serve as the eigenvectors

However, because the variances are equal, any rotated set of axes (i.e., any rotation of the standard basis) would be valid principal components. In other words, when the variances are equal, there is no single dominant direction, so the principal components are not uniquely defined.

2 Q2

Part 1: Maximization of f^TCf under the constraint $f \perp e$

We are given the following setup:

- A set of N vectors $X = \{x_1, x_2, \ldots, x_N\}$ in $\mathbb{R}^d$, with average vector $\bar{x}$.
- The covariance matrix C of the vectors in X.
- The direction e that maximizes e^TCe, which is the eigenvector corresponding to the highest eigenvalue λ_1 of C.

We want to find the vector f that maximizes f^TCf, subject to the constraint that f is orthogonal to e (i.e., $f^Te = 0$) and that $\|f\| = 1$ (i.e., f is a unit vector).

Since C is a symmetric matrix (as proved in Q1), by spectral decomposition it can be written as:

$$C = \sum_{i=1}^{d} \lambda_i v_i v_i^T$$

where λ_i are the eigenvalues of C, and v_i are the corresponding orthonormal eigenvectors.

We know that $e = v_1$, the eigenvector corresponding to the largest eigenvalue λ_1. To maximize f^TCf, we will express f as a linear combination of the eigenvectors of C:

$$f = \sum_{i=1}^{d} \alpha_i v_i$$

Since f is orthogonal to $e = v_1$, this imposes the constraint that $\alpha_1 = 0$. Therefore, f must lie in the subspace spanned by the remaining eigenvectors $v_2, v_3, \ldots, v_d$.

Thus, we can write:

$$f = \sum_{i=2}^{d} \alpha_i v_i$$

Now, let's calculate f^TCf:

$$f^TCf = \left(\sum_{i=2}^{d} \alpha_i v_i\right)^T \left(\sum_{j=1}^{d} \lambda_j v_j v_j^T\right) \left(\sum_{k=2}^{d} \alpha_k v_k\right)$$

Using the orthogonality of the eigenvectors v_i (i.e., $v_i^T v_j = \delta_{ij}$, where δ_{ij} is the Kronecker delta), this simplifies to:

$$f^TCf = \sum_{i=2}^{d} \lambda_i \alpha_i^2$$

Our goal is to maximize $\sum_{i=2}^{d} \lambda_i \alpha_i^2$ subject to the constraint $\sum_{i=2}^{d} \alpha_i^2 = 1$, which ensures that f remains a unit vector.

To maximize this sum, we should choose f to align with the eigenvector corresponding to the largest eigenvalue among $\lambda_2, \lambda_3, \ldots, \lambda_d$. Let say λ_2 is the second-highest eigenvalue (other than λ_1, which is the largest among all), the direction f that maximizes f^TCf is the eigenvector v_2 corresponding to λ_2.

Thus, the direction f that maximizes f^TCf, subject to $f \perp e$, is the eigenvector corresponding to the second-highest eigenvalue λ_2.

Part 2: Extend to the vector g perpendicular to both e and f

Next, we extend this reasoning to find the unit vector g that maximizes $g^T C g$, subject to the constraints that g is perpendicular to both $e = v_1$ and $f = v_2$.

By similar reasoning, we express g as a linear combination of the remaining eigenvectors:

$$g = \sum_{i=3}^{d} \beta_i v_i$$

and we want to maximize:

$$g^T C g = \sum_{i=3}^{d} \lambda_i \beta_i^2$$

subject to $\sum_{i=3}^{d} \beta_i^2 = 1$.

The maximum is achieved when g is aligned with the eigenvector corresponding to the largest remaining eigenvalue, which is λ_3. Thus, the direction g that maximizes $g^T C g$, subject to $g \perp e$ and $g \perp f$, is the eigenvector v_3 corresponding to λ_3, which is the third largest among all the eigenvalues.

Spectral Decomposition Proof for a Symmetric Matrix
Let C be a symmetric matrix of size $d \times d$, such that $C = C^T$.
The spectral decomposition theorem states that C can be expressed as:
$C = \sum_{i=1}^{d} \lambda_i v_i v_i^T$
where λ_i are the eigenvalues of C, and v_i are the corresponding orthonormal eigenvectors.

Since C is symmetric, the Spectral Theorem guarantees that:
- The eigenvalues of C are real.
- The eigenvectors of C are orthonormal, i.e., $v_i^T v_j = 0$ for $i \neq j$ and $v_i^T v_i = 1$.

Therefore, C can be diagonalized as:
$C = V \Lambda V^T$
where V is the matrix of eigenvectors, and Λ is the diagonal matrix of eigenvalues.

The matrix equation $C = V \Lambda V^T$ can be rewritten as:
$C = \sum_{i=1}^{d} \lambda_i v_i v_i^T$
Each term $\lambda_i v_i v_i^T$ represents a rank-1 matrix that projects onto the eigenvector v_i, scaled by λ_i.

Since the eigenvectors are orthonormal, $v_i^T v_j = 0$ for $i \neq j$ and $v_i^T v_i = 1$.
This ensures that the matrix C can be reconstructed from the sum of these outer products.

3 Q3

3.1 (a)

Using Singular Value Decomposition (SVD), we can write any $m \times n$ matrix **A** as:

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T \tag{1}$$

where: - **U** is an $m \times m$ orthogonal matrix (whose columns are the left singular vectors of **A**), - is an $m \times n$ diagonal matrix with the singular values σ_i of **A** on the diagonal, - **V** is an $n \times n$ orthogonal matrix (whose columns are the right singular vectors of **A**).

Thus, we can express **A** as a sum over its singular values and corresponding singular vectors:

$$\mathbf{A} = \sum_i \sigma_i u_i v_i^T \tag{2}$$

where: - u_i is the i-th column of **U** (an $m \times 1$ orthonormal vector), - v_i is the i-th column of **V** (an $n \times 1$ orthonormal vector), - σ_i are the singular values of **A**.

Now, consider the product $\mathbf{A}\mathbf{A}^T$:

$$\mathbf{A}\mathbf{A}^T = \left(\sum_i \sigma_i u_i v_i^T\right)\left(\sum_j \sigma_j v_j u_j^T\right) \tag{3}$$

Since $v_i^T v_j = \delta_{ij}$ (the Kronecker delta), the above expression simplifies to:

$$\mathbf{A}\mathbf{A}^T = \sum_i \sigma_i^2 u_i u_i^T \tag{4}$$

Similarly, we can write $\mathbf{A}^T\mathbf{A}$ as:

$$\mathbf{A}^T\mathbf{A} = \sum_i \sigma_i^2 v_i v_i^T \tag{5}$$

We know that the eigendecomposition of any symmetric matrix **B** can be written as:

$$\mathbf{B} = \mathbf{U}\mathbf{U}^T = \sum_i \lambda_i u_i u_i^T \tag{6}$$

By comparing eq(6) and eq(4), the non-zero eigenvalues of both $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$ will be σ_i^2. Therefore, we can conclude that the non-zero singular values σ_i of **A** are the square roots of the eigenvalues of both $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$.

3.2 (b)

The Frobenius norm of a matrix **A** is defined as:

$$||\mathbf{A}||_F = \sqrt{\sum_{ij} |a_{ij}|^2} = \sqrt{\mathrm{Tr}(\mathbf{A}\mathbf{A}^T)} \tag{7}$$

This equation is valid only if all elements of $\mathbf{A}$ are real. If $\mathbf{A}$ is complex, we need to take the conjugate transpose instead of the regular transpose. In that case, the Frobenius norm becomes:

$$||\mathbf{A}||_F = \sqrt{\text{Tr}(\mathbf{A}\mathbf{A}^\dagger)} \tag{8}$$

Using the Singular Value Decomposition (SVD), we can express the Frobenius norm as follows. Given $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$, where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and Σ is the diagonal matrix containing the singular values of $\mathbf{A}$, we can calculate the Frobenius norm:

$$||\mathbf{A}||_F = \sqrt{\text{Tr}(\mathbf{U}\Sigma\mathbf{V}^T\mathbf{V}\Sigma\mathbf{U}^T)} \tag{9}$$

Since $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ (orthogonality of $\mathbf{V}$), this simplifies to:

$$||\mathbf{A}||_F = \sqrt{\text{Tr}(\mathbf{U}\Sigma^2\mathbf{U}^T)} = \sqrt{\text{Tr}(\Sigma^2)} \tag{10}$$

The trace of Σ^2 is simply the sum of the squares of the singular values (or eigenvalues, for symmetric matrices) of $\mathbf{A}$. Therefore, we have:

$$||\mathbf{A}||_F = \sqrt{\sum_i \sigma_i^2} \tag{11}$$

where σ_i are the singular values of $\mathbf{A}$. Thus, the Frobenius norm is the square root of the sum of the squares of the singular values (or the eigenvalues in the case of symmetric matrices).

3.3 (c)

[The testing for error in MATLAB is provided in `./3/code/myMainScript.m`]

The student's approach to computing the SVD of a matrix A using the eigendecomposition of A^TA and AA^T is conceptually sound but prone to subtle issues due to differences between eigendecomposition and SVD. The key problems likely lie in the following aspects:

1. Sign Ambiguity of Eigenvectors

Eigenvectors of a matrix are not unique in their direction — they are defined up to a sign. If $\mathbf{v}$ is an eigenvector, then $-\mathbf{v}$ is also a valid eigenvector with the same eigenvalue. In the SVD, the matrices U and V (which consist of the left and right singular vectors) must have specific signs that align such that:

$$USV^T = A.$$

However, when computing the eigendecomposition, MATLAB's `eig` function may return eigenvectors with arbitrary signs, meaning that the signs of the eigenvectors of A^TA (for V) and AA^T (for U) may not match in a way that reconstructs A correctly when multiplying USV^T.

2. Ordering of Singular Values and Eigenvectors

Another potential issue could be the ordering of the singular values and their corresponding eigenvectors. In the SVD, the singular values (and their corresponding singular vectors) are typically arranged in descending order. However, in eigendecomposition, MATLAB might not always return the eigenvalues and eigenvectors in the same order, which could lead to a mismatch between the singular values and the corresponding singular vectors in U and V.

Solution: **Post-processing the Eigenvectors**

To rectify this error, the student needs to ensure the following after computing the eigenvectors:

1. Fix the Signs of the Eigenvectors

The student should compare the eigenvectors of AA^T (for U) and A^TA (for V) to ensure that their signs are consistent. One way to check is to compute $A\mathbf{v}_i$ for each right singular vector $\mathbf{v}_i$ and ensure that it aligns with the corresponding left singular vector $\mathbf{u}_i$. If the signs do not match, the student should flip the sign of the corresponding eigenvector.

2. Sort the Eigenvalues and Eigenvectors

The student should sort the eigenvalues of A^TA (or AA^T) in descending order and reorder the corresponding eigenvectors accordingly to ensure that the singular values and vectors match.

3.4 (d)

The question has defined:

$$\mathbf{P} = \mathbf{A}^T\mathbf{A} \quad \& \quad \mathbf{Q} = \mathbf{A}\mathbf{A}^T \tag{12}$$

(i)

Let $\mathbf{y}$ be an $n \times 1$ vector. Using the Singular Value Decomposition (SVD) of $\mathbf{P}$, we can express the quadratic form as:

$$\mathbf{y}^T\mathbf{P}\mathbf{y} = \mathbf{y}^T \left(\sum_i \sigma_i^2 \mathbf{u}_i \mathbf{u}_i^T \right) \mathbf{y} = \sum_i \sigma_i^2 \mathbf{y}^T \mathbf{u}_i \mathbf{u}_i^T \mathbf{y} \tag{13}$$

Since $\mathbf{y}^T\mathbf{u}_i$ is a scalar, we simplify this to:

$$\mathbf{y}^T\mathbf{P}\mathbf{y} = \sum_i \sigma_i^2 (\mathbf{u}_i^T \mathbf{y})^2 \tag{14}$$

All terms in the summation are squared, so it follows that:

$$\mathbf{y}^T\mathbf{P}\mathbf{y} \geq 0 \tag{15}$$

Similarly, let $\mathbf{z}$ be an $m \times 1$ vector. Using SVD on $\mathbf{Q}$, we can write:

$$\mathbf{z}^T\mathbf{Q}\mathbf{z} = \mathbf{z}^T \left(\sum_i \sigma_i^2 \mathbf{v}_i \mathbf{v}_i^T \right) \mathbf{z} = \sum_i \sigma_i^2 \mathbf{z}^T \mathbf{v}_i \mathbf{v}_i^T \mathbf{z} \tag{16}$$

Simplifying as before, we get:

$$\mathbf{z}^T\mathbf{Q}\mathbf{z} = \sum_i \sigma_i^2 (\mathbf{v}_i^T \mathbf{z})^2 \tag{17}$$

Again, since all terms in the summation are squared, we conclude that:

$$\mathbf{z}^T\mathbf{Q}\mathbf{z} \geq 0 \tag{18}$$

Eigenvalues of P:

If λ is an eigenvalue of P with eigenvector **u**:

$$P\mathbf{u} = \lambda\mathbf{u}$$

Then:

$$\mathbf{u}^T P\mathbf{u} = \lambda\mathbf{u}^T\mathbf{u} \geq 0$$

Since $\mathbf{u}^T\mathbf{u} > 0$ for non-zero **u**, it follows that $\lambda \geq 0$.

Eigenvalues of Q:

If μ is an eigenvalue of Q with eigenvector **v**:

$$Q\mathbf{v} = \mu\mathbf{v}$$

Then:

$$\mathbf{v}^T Q\mathbf{v} = \mu\mathbf{v}^T\mathbf{v} \geq 0$$

Again, since $\mathbf{v}^T\mathbf{v} > 0$ for non-zero **v**, it follows that $\mu \geq 0$.

(ii)

The problem defines the following eigenvalue equation:

$$\mathbf{P}\mathbf{u} = \lambda\mathbf{u} \tag{19}$$

which implies:

$$\mathbf{A}^T\mathbf{A}\mathbf{u} = \lambda\mathbf{u} \tag{20}$$

Now, multiplying both sides by **A**:

$$\mathbf{A}\mathbf{A}^T\mathbf{A}\mathbf{u} = \lambda\mathbf{A}\mathbf{u} \quad \Rightarrow \quad \mathbf{Q}(\mathbf{A}\mathbf{u}) = \lambda(\mathbf{A}\mathbf{u}) \tag{21}$$

Thus, **Au** is an eigenvector of **Q** with eigenvalue λ.

Similarly, the problem also defines:

$$\mathbf{Q}\mathbf{v} = \mu\mathbf{v} \tag{22}$$

which implies:

$$\mathbf{A}\mathbf{A}^T\mathbf{v} = \mu\mathbf{v} \tag{23}$$

Now, multiplying both sides by $\mathbf{A}^T$:

$$\mathbf{A}^T\mathbf{A}\mathbf{A}^T\mathbf{v} = \mu\mathbf{A}^T\mathbf{v} \quad \Rightarrow \quad \mathbf{P}(\mathbf{A}^T\mathbf{v}) = \mu(\mathbf{A}^T\mathbf{v}) \tag{24}$$

Hence, $\mathbf{A}^T\mathbf{v}$ is an eigenvector of $\mathbf{P}$ with eigenvalue μ.

Number of Elements in u and **v**:

The vector **u** is in the domain of P and thus has the same dimension as the input space of A (which maps to the domain of Q).

The vector **v** is in the domain of Q and has the same dimension as the output space of A^T (which maps back to the domain of P).

Thus, both **u** and **v** should have the same number of elements as the number of rows of A.

(iii)

Let v_i be an eigenvector of **Q**, and we define u_i as:

$$u_i \triangleq \frac{\mathbf{A}^T v_i}{||\mathbf{A}^T v_i||_2} \tag{25}$$

Now, multiplying both sides by **A**, we get:

$$\mathbf{A}u_i = \frac{\mathbf{A}\mathbf{A}^T v_i}{||\mathbf{A}^T v_i||_2} = \frac{\mathbf{Q}v_i}{||\mathbf{A}^T v_i||_2} = \frac{\mu v_i}{||\mathbf{A}^T v_i||_2} \tag{26}$$

This simplifies to:

$$\mathbf{A}u_i = \left(\frac{\mu}{||\mathbf{A}^T v_i||_2}\right) v_i \tag{27}$$

Thus, we define:

$$\gamma_i = \left(\frac{\mu}{||\mathbf{A}^T v_i||_2}\right) \tag{28}$$

where γ_i will be non-negative since μ is non-negative, as previously shown.

3.4.1 (iv)

Proof of Singular Value Decomposition

To show that $A = U\Gamma V^T$, where Γ is a diagonal matrix containing the non-negative singular values

Let:

$$U = [\mathbf{u}_1|\mathbf{u}_2|\mathbf{u}_3|\ldots|\mathbf{u}_m]$$

be a matrix formed by the eigenvectors of $P = A^T A$ corresponding to its non-negative eigenvalues. Let:

$$V = [\mathbf{v}_1|\mathbf{v}_2|\mathbf{v}_3|\ldots|\mathbf{v}_m]$$

be a matrix formed by the eigenvectors of $Q = AA^T$ corresponding to its non-negative eigenvalues.

Assume λ_i is the eigenvalue corresponding to the eigenvector $\mathbf{u}_i$ for matrix P, and let $\gamma_i = \sqrt{\lambda_i}$ be the singular value of A. Since λ_i is non-negative, γ_i is also non-negative.

Using the definitions of P and Q:

- From the definition of P:

$$P\mathbf{u}_i = A^T A\mathbf{u}_i = \lambda_i \mathbf{u}_i$$

- From the definition of Q:

$$Q\mathbf{v}_i = AA^T\mathbf{v}_i = \mu_i \mathbf{v}_i$$

  where μ_i corresponds to the eigenvalue of Q.

We can express A in terms of U, V, and Γ:

1. We know that:

$$A\mathbf{u}_i = \gamma_i \mathbf{v}_i$$

   for $i = 1, 2, \ldots, m$.

2. Therefore, for all i, we can write:

$$AU = \begin{bmatrix} A\mathbf{u}_1 & A\mathbf{u}_2 & \ldots & A\mathbf{u}_m \end{bmatrix} = \begin{bmatrix} \gamma_1\mathbf{v}_1 & \gamma_2\mathbf{v}_2 & \ldots & \gamma_m\mathbf{v}_m \end{bmatrix}$$

Constructing Γ

Define Γ as:

$$\Gamma = \text{diag}(\gamma_1, \gamma_2, \ldots, \gamma_m)$$

Final Expression

Combining all these results, we have:

$$AU = U\Gamma V^T$$

Thus, we can write:

$$A = U\Gamma V^T$$

4 Q4

The codes are provided in 2 different files (`./4/code/myMainScript_ORL.m` and `./4/code/myMainScript_Yale.m`) for modularity and the fact that the datasets have different structures and naming conventions.

4.1 ORL Dataset

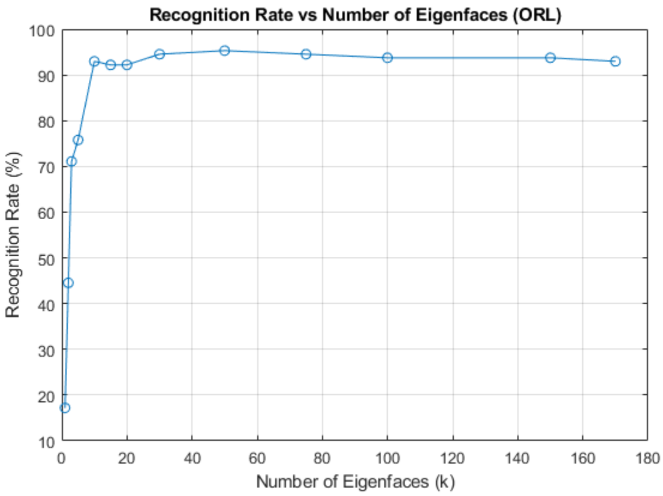

Figure 1: Recognition Rates for ORL Datasets using `eigs`

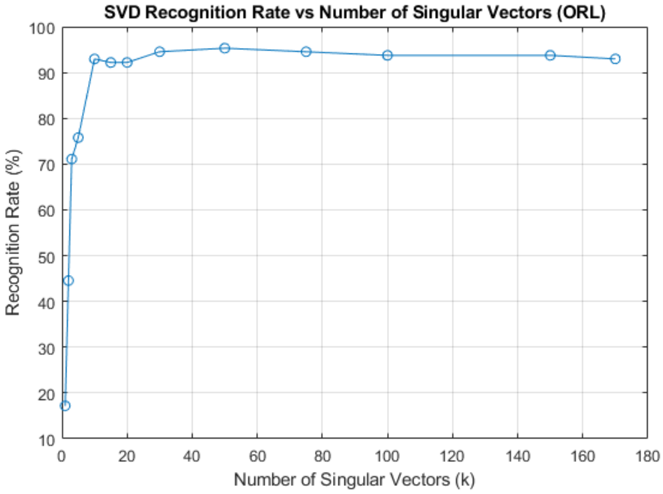

Figure 2: Recognition Rates for ORL Datasets using `svd`

The results of both the methods (using `svd` and `eig`) is nearly if not exactly same. It's observed that just after **10** Eigenfaces, the recognition rate results starts to accumulate showing that for efficient use, the recognition can be done using just **10** EFs.

Face Reconstruction

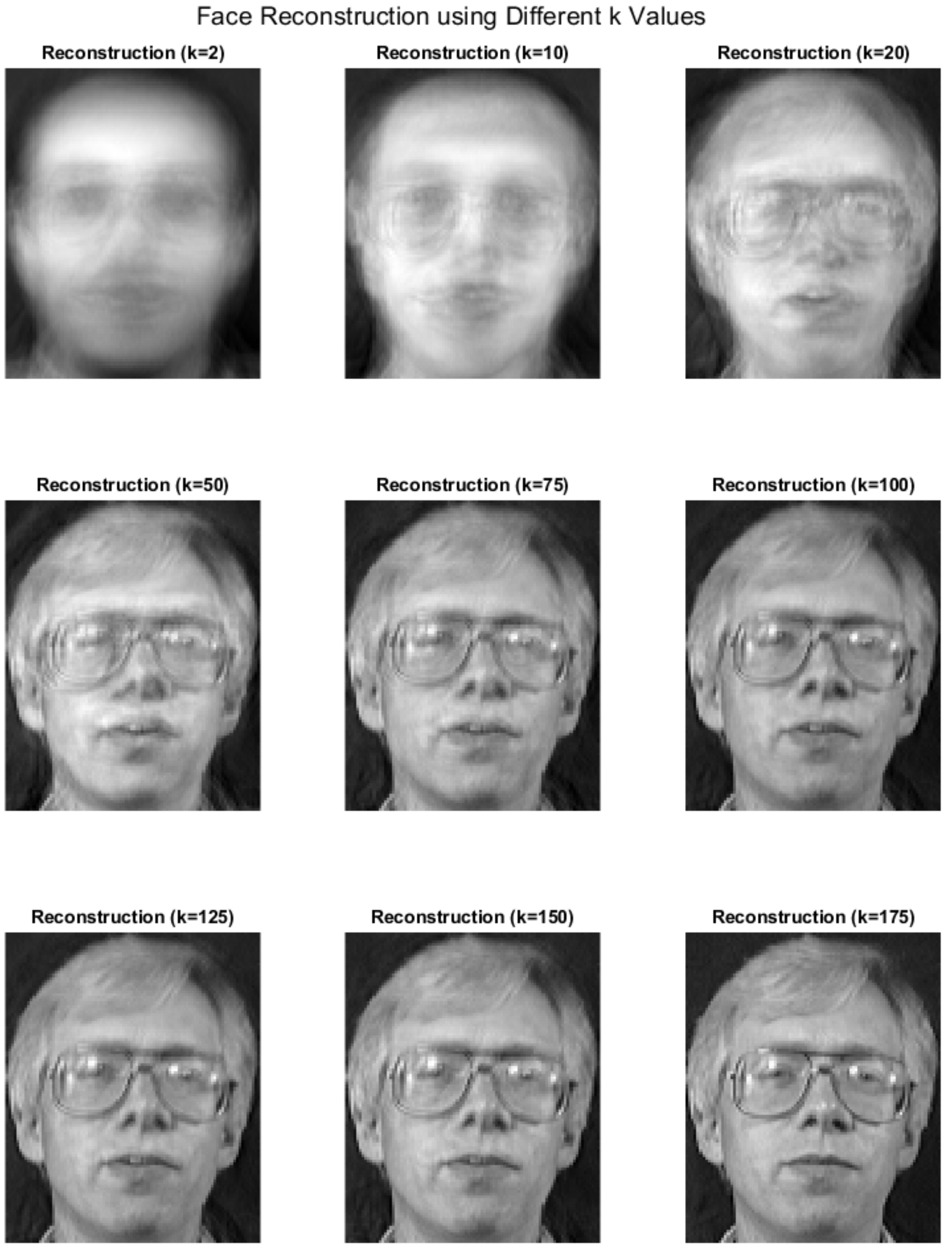

Figure 3: Face Reconstruction for different k values

Top 25 largest Eigenvectors

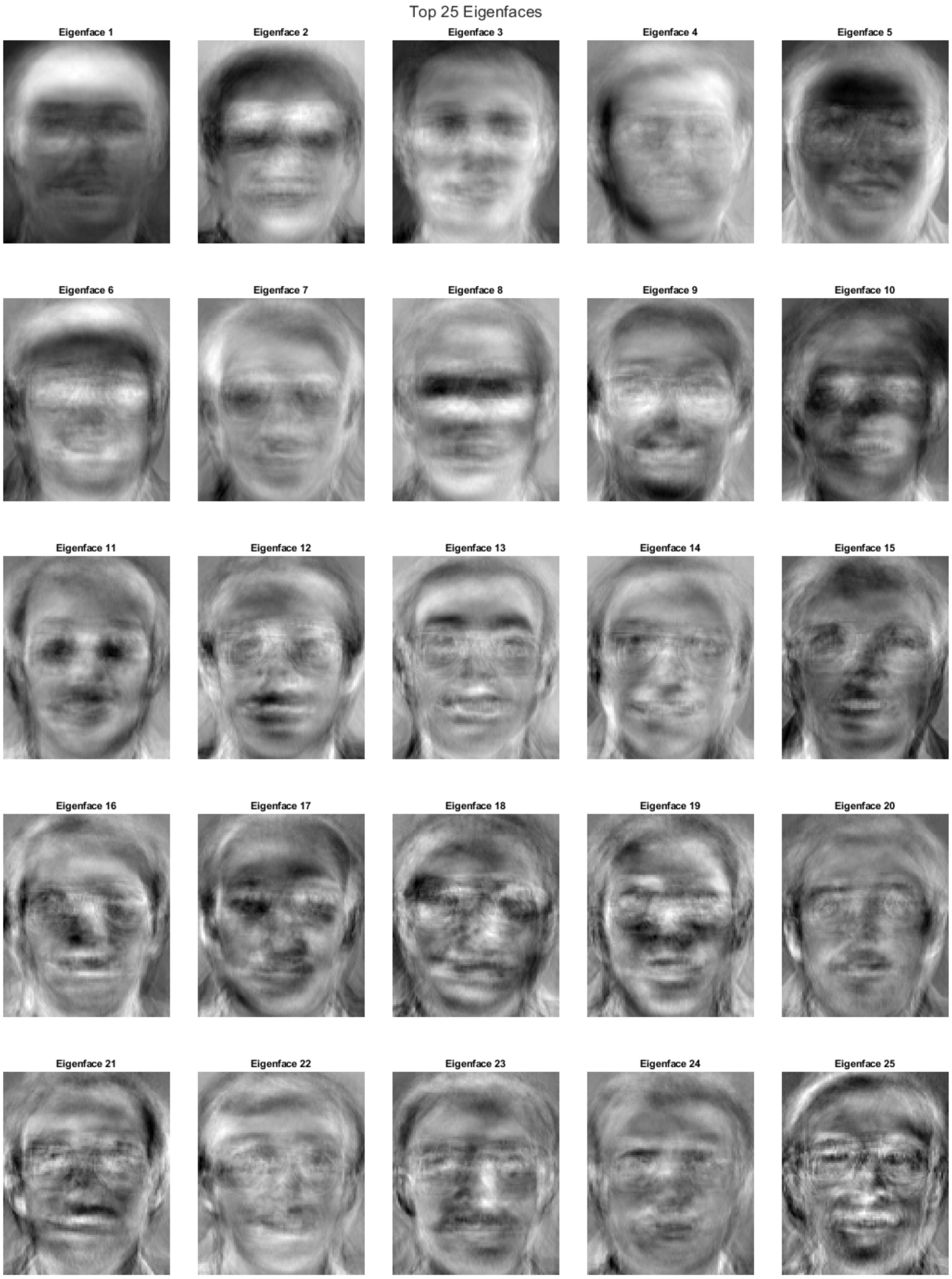

Figure 4: Eigenvectors corresponding to Top 25 largest Eigenvalues

4.2 Yale Dataset

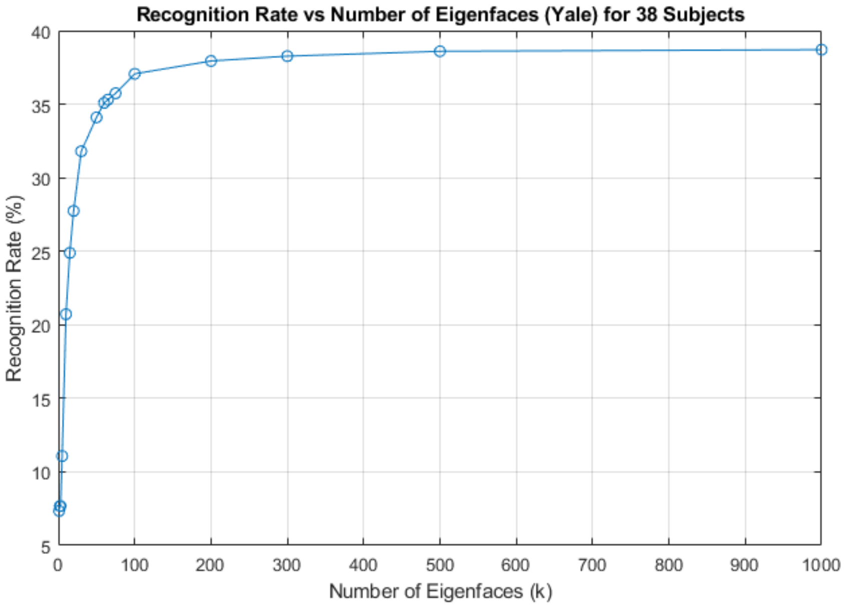

Figure 5: Recognition Rates for Yale Dataset

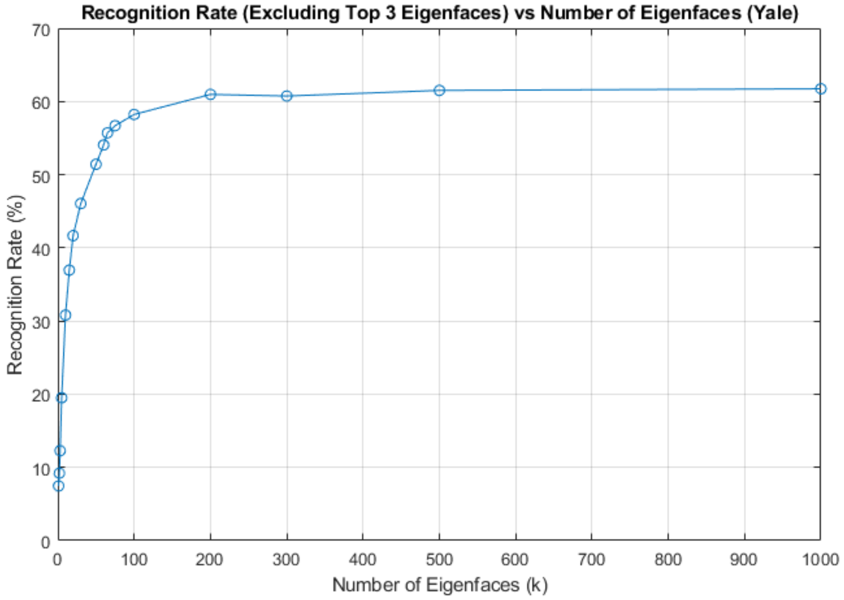

Figure 6: Recognition Rates for Yale Datasets (excluding top 3)

In face recognition tasks, varying lighting conditions can significantly impact the performance of recognition algorithms. The first figure (Figure 5) illustrates the recognition rates for the Yale dataset.

However, as shown in the second figure (Figure 6), which presents the results after excluding the top three eigenvectors, there is a notable improvement in recognition rates. This suggests that by mitigating the impact of these **top specific eigenvectors**, which may be **overly sensitive to lighting variations**, we can achieve more robust recognition performance across different lighting conditions.

5 Q5

Implementation and comparisons provided in `./5/code/myMainScript.m` sub-folder.

To address the problem of recognizing images of people who were not part of the training set and detecting when there is no matching identity, we can use a **threshold-based method**. This mechanism will report an ”unknown identity” if the distance between the test projection and all training projections is above a predefined threshold.

The idea is to compute the distance between the test image projection and the projections of the training images for each subject. If the minimum distance exceeds the threshold, we classify the test image as having no matching identity.

Key Steps would include:

- **Compute Euclidean distances** between the test image projection and all training image projections.
- Define a **threshold** based on distance metrics from the training set (can be auto-tuned using regression methods, in code, it is manually defined).
- If the minimum distance for a test image is greater than the threshold, report ”unknown identity.”
- **Count** false positives (incorrectly identifying an ”unknown” person as a known identity) and false negatives (incorrectly identifying a known identity as ”unknown”).